How dCache Namespace Works

- View from database performance
  - PNFS
  - Chimera

How PNFS works

- All files and directories have unique PNFSID, which is a unique key in the table.
- All meta data associated with files and directories also have PNFSIDs (differ from PNFSID of associated files)
- The data associated with PNFSIDs are stored as “blob”.
- “blob” is used like hash.
    - “blob” not only include meta information (sometimes) but also include PNFSIDs
        - “blob” needs to be decoded to link to the next PNFSIDs → CPU Expensive
        - Size of “blob” is limited.

Content of directory data blob

- Data Blob of associated with PNFID of directory contains five PNFSIDs (at least)
  - PNFSID of itself
  - PNFSID of metadata PNFSIDs
  - PNFSID of parent directory
  - Two PNFSIDs
    - The data blob of each of two PNFSIDs contains list of all PNFSIDs of files and and sub-directories within this directory if the number is relatively small (~100) If not, it contains

Dcache PNFS example

- srm://dcsrm.usatlas.bnl.gov/pnfs/usatlas.bnl.gov/BNLT1D0/data08_cos/RAW/data08_cos.00072854.physics_HLT_Cosmics_MU3.daq.RAW.o3/daq.CTPRPCTGC.0072854.physics.HLT_Cosmics_MU3.LB0026.SFO-5._0001.data
- BNLT1D0 → PNFSID of “BNLT1D0” → data blob contains the two PNFSIDs.
  - Data blobs for these two PNFSIDs contain the list of many PNFSIDs.
    - Data for these PNFSIDs from these two list contain the name of files and/or directories and associated PNFSID, one of which corresponds to “data08_cos” sub directory
- data08_cos → Repeat the same process to get PNFSID of “RAW” sub directory.
- RAW → Again, repeat the process to get PNFSID of “data08_cos.....03” sub directory. However, these is a difference due to the size limit in data blob. Since “RAW” contains many subdirectories (~1k), the list of PNFSIDs can not contain all subdirectories. As a result, it uses the list of the list, resulting more queries.
- data08_cos.00072854.physics_HLT_Cosmics_MU3.daq.RAW.o3 → Again, repeat the same process to get PNFSID of
  the file, daq.CTPRPCTGC.0072854.physics.HLT_Cosmics_MU3.LB0026.SFO-5._0001.data
  But, also, get meta data for this file, which have different PNFSIDs. (yet more quries)

PNFS Summary

- “ls” of directory is very expensive in PNFS due to the requirment of (many) SQL queries + decodeing of many blobs
- During the high load time, it is CPU limited. Decoding of blob is expensive!
- Although PNFS database design is highly limited, there is one nice feature. That is that PNFS deamon catches the information for subsequent requests.

Chimera

- It does not use “blob” data → no decoding of “blob”
    - one SQL query will get all files in one directories vs many SQL queries to get all files in PNFSD
        - “ls” of directories in Chimera will be much faster than in PNFS
- Very similar in design to LFC (another psedo file system)
- Look up by the multiple clients should work faster due to the non-blob-decoding. In PNFS, blob-decoding acts like the table lock.
- It does not seem to catch the previous SQL lookup. As a result, it requires the similar number of real SQL queires to get the specific file information as PNFD.

Chimera Schema

```
                iparent                |      iname      |               ipnfsid
---------------------------------------+-----------------+--------------------------------------
...
...
 000000000000000000000000000000000000 | pnfs            | 000039DCBE4B7CD144C386DF6DC060C238AA
 000039DCBE4B7CD144C386DF6DC060C238AA | .               | 000039DCBE4B7CD144C386DF6DC060C238AA
 000039DCBE4B7CD144C386DF6DC060C238AA | ..              | 000000000000000000000000000000000000
 000039DCBE4B7CD144C386DF6DC060C238AA | usatlas.bnl.gov | 0000EDCFFAA3B6504CEA812425A628EF5515
 0000EDCFFAA3B6504CEA812425A628EF5515 | .               | 0000EDCFFAA3B6504CEA812425A628EF5515
 0000EDCFFAA3B6504CEA812425A628EF5515 | ..              | 000039DCBE4B7CD144C386DF6DC060C238AA
...
 0000EDCFFAA3B6504CEA812425A628EF5515 | data            | 000026B93E15908E4D188943A429A13B6E9D
 000026B93E15908E4D188943A429A13B6E9D | .               | 000026B93E15908E4D188943A429A13B6E9D
 000026B93E15908E4D188943A429A13B6E9D | ..              | 0000EDCFFAA3B6504CEA812425A628EF5515
...
 000026B93E15908E4D188943A429A13B6E9D | iriswu          | 0000F4DDB2480AE74ACBB5773C210EE39B2C
 0000F4DDB2480AE74ACBB5773C210EE39B2C | .               | 0000F4DDB2480AE74ACBB5773C210EE39B2C
 0000F4DDB2480AE74ACBB5773C210EE39B2C | ..              | 000026B93E15908E4D188943A429A13B6E9D
```

```
                iparent                | iname  |               ipnfsid
---------------------------------------+--------+--------------------------------------
 000026B93E15908E4D188943A429A13B6E9D | iriswu | 0000F4DDB2480AE74ACBB5773C210EE39B2C
```

```
                iparent                |    iname     |               ipnfsid
---------------------------------------+--------------+--------------------------------------
 0000F4DDB2480AE74ACBB5773C210EE39B2C | sub1_dir8999 | 0000AE61254043B14B85B417FDB0FEAEB6CA
```

```
                iparent                |    iname     |               ipnfsid
---------------------------------------+--------------+--------------------------------------
 0000AE61254043B14B85B417FDB0FEAEB6CA | testfile9901 | 00001E668A6A7D3E4760932F8D43EBAFB52D
```

Comparison of PNFS and Chimera from Datatabase trace

- Test setup.
  - /A/B/C/File.i   i=0..10000
- PNFS
  - ~15 SQLs
- Chimera
  - ~12 SQLs
- Single thread performance of “ls /A/B/C/File.i” shows Chimera being 27% improvement.